LA-UR-18-29597
Approved for public release; distribution is unlimited.

Title: Developing a nonparametric functional calibration framework for physics-based models

Author(s): Chodora, Evan Christopher

Intended for: Report

Issued: 2018-10-10

Developing a nonparametric functional calibration framework for physics-based models

Evan Chodora, W-13

1. Introduction

Due to the cost, complexity, or difficulty in solely carrying out experiments in the effort to better understand physical phenomenon, researchers and engineers often work towards the development of computer models to represent the systems under analysis. The importance of these models in accurately representing the systems that they are designed to emulate has become a focus in numerous fields, including medicine, energy, defense, and manufacturing.

The goal of this effort to improve computer model representations has its foundation in determining the best values of the parameters that control the model output so that it most accurately matches the real-world response of the system. While many parameters remain fairly constant throughout the operational domain of the system, there are parameters that vary significantly under different control inputs. Parameters such as thermal conductivity changing with temperature or density changing with pressure can have considerable impacts on the measured output of a system. These functional variations may be known to the calibrator ahead of time due to a known mathematical or measured relationship and can then be implemented directly into the model. If the relationship is predominately linear, for example, the slope and intercept of the line can be calibrated as two separate parameters to represent that relationship. If the functional form of the relationship is unknown, however, this becomes difficult to incorporate.

The aim of this work is to solve this difficulty and provide a framework for both determining which model parameters may have functional dependence and then calibrating the unknown functional responses of physics-based model parameters as smooth functions of the operational control inputs.

2. Functional parameter sensitivity analysis

One of the biggest challenges in functional calibration is deciding which of many parameters in a computer model may be functionally dependent on the control inputs. The additional computational cost over calibration where all the parameters are scalars makes the determination of which, if any, model parameters should be calibrated using nonparametric, functional techniques a necessity.

One way to conduct this sensitivity analysis that is proposed in this work is to calibrate the model using scalar parameters over small portions of the domain on which dependence is expected and analyzing how the mean of the parameter posterior distributions vary for each portion of the domain. This staged process allows for the determination of which parameters need to vary more throughout the domain in order to best agree with experimental results. This approach uses Bayesian model calibration methods to determine the most likely values (the posterior distributions) for the model parameters through the integration of measured data and the computer model outputs (Higdon, Kennedy, Cavendish, Cafeo, & Ryne, 2004) (Williams, et al., 2006).

Once the most likely values of the calibration parameters are determined, Equation (1) below represents the relative variation in the calibration parameter θ_j due to the change in the input domain range.

$$V_j = \frac{\sqrt{\frac{\sum_{i=1}^{n}\left(\mu_j^i - \bar{\mu}_j\right)^2}{n-1}}}{\frac{\sum_{i=1}^{n}\mu_j^i}{n}} \tag{1}$$

Where μ_j^i represents the posterior mean of θ_j calculated using domain range i, n is the number of ranges the domain was separated into, and $\bar{\mu}_j$ is the mean of all the calibrated values for θ_j used in the calculation of the variation to normalize the standard deviation. This value can then serve as a relative indicator as to how much the most likely value of θ_j is dependent on the control input range.

2.1. Demonstration on an algebraic problem

To demonstrate this approach, an algebraic problem is devised as an example that consists of three total parameters. Two of these parameters (θ_1 and θ_2) are constant and one (τ) is functionally dependent on the input, x. The equation of the model is shown below in Equation (2).

$$y = \theta_1 x^2 + \theta_2 x + \tau \tag{2}$$

The parameter nominal values correspond to θ_1 = 2.5, θ_2 = 3.0, and $\tau = 2\sqrt{x}$. Figure 1 below shows the output of this model as the value of the input is varied between 0 and 1.

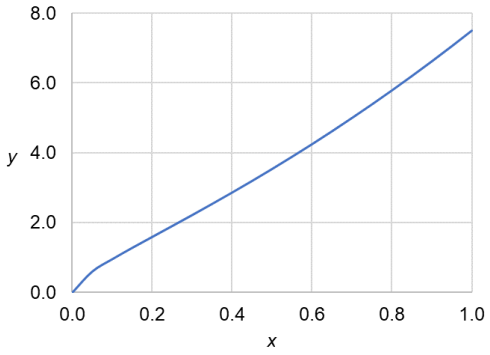

Figure 1. Output from the algebraic model problem as a function of the value of x.

Based on the model, the input domain was broken up into three distinct ranges: [0.00, 0.30], [0.35, 0.65], and [0.70, 1.00]. The model is then run over those input ranges to collect three vectors of y values corresponding to each range and representing the measured response of the system.

Using the measured responses, three Bayesian calibration runs were carried out for each of the input ranges using the Dakota software developed by Sandia National Laboratories (Adams, et al., 2014). Dakota integrates the QUESO (Quantification of Uncertainty for Estimation Simulation and Optimization) C++ library developed by the Center for Predictive Engineering and Computational Science at the University of Texas at Austin for Bayesian analysis (Estacio-Hiroms,

Prudencio, Malaya, Vohra, & McDougall, 2016). The results for each run are shown in Table 1 below with the mean and standard deviation of each model parameter.

Table 1. Results from calibration over each input range

Input Range	θ_1		θ_2		τ	
	$\bar{\mu}$	σ	$\bar{\mu}$	σ	$\bar{\mu}$	σ
1	2.384	1.398	2.607	0.752	0.891	0.399
2	2.900	1.302	2.737	0.779	2.679	0.209
3	3.553	1.070	3.602	0.284	2.934	0.050

Using the data in Table 1, the relative variation for each parameter is calculated using Equation (1) and shown below in Figure 2. This plot shows that τ has over 2.5 times the relative variation compared to either of the constant parameters in the model. This provides a probable indication that this parameter may be suited to being treated as functionally dependent over the selected input domain.

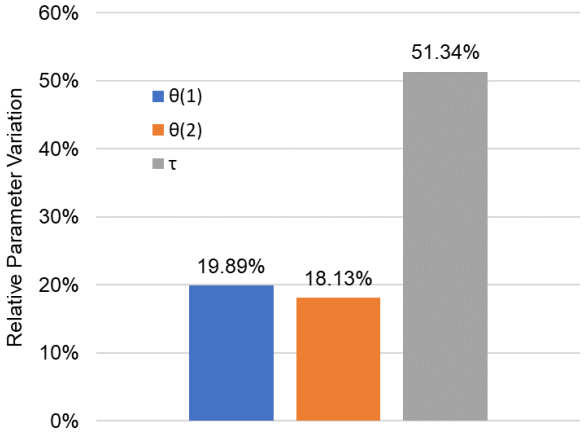

Figure 2. Relative variation of the three parameters due to calibration over three control input ranges.

It is important to note, however, that the values of the calibrated parameters in this approach for sensitivity analysis are not necessarily important or accurate as they do not use information about the entire measured response range. This is especially true when one or more parameter is functionally dependent on the input domain and the parameters may compensate for each other based on their individual sensitivities over each region of the domain.

2.2. Comparison to three scalar parameters

To compare the relative variation approach between a model with a functional parameter to one with all constant parameters, the model is modified as shown in Equation (3).

$$y = \theta_1 x^2 + \theta_2 x + \tau\sqrt{x} \tag{3}$$

The parameter nominal values correspond to $\theta_1 = 2.5$, $\theta_2 = 3.0$, and $\tau = 2.0$ allowing the same measured output response vectors to be used. The same Bayesian calibration method over the three input domains was repeated on the model in Equation (3) and the results are shown below in Table 2.

Table 2. Results from calibration over each input range with a scalar parameter model

Input Range	θ_1		θ_2		τ	
	$\bar{\mu}$	σ	$\bar{\mu}$	σ	$\bar{\mu}$	σ
1	2.072	1.038	2.236	1.073	1.907	0.695
2	2.381	1.027	2.681	0.809	2.154	0.624
3	2.519	0.937	2.834	0.751	2.250	0.660

Using the data in Table 2, the relative variation for each parameter is calculated using Equation (1) and shown below in Figure 3. The results of this calculation would appear to indicate there is not a strong reason to believe any of the three parameters would benefit from being treated as functionally dependent relative to the others.

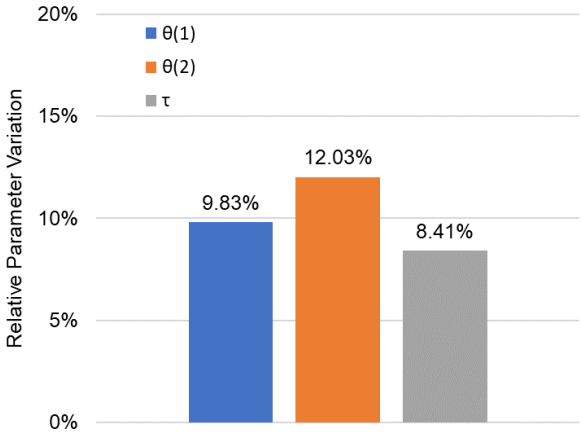

Figure 3. Relative variation of the three scalar parameters due to calibration over three control input ranges.

3. Nonparametric calibration method

Much of the work in the last two decades or so have traditionally considered calibration parameters as fixed, scalar values that are constant throughout the operational input domain of the model. The work of (Kennedy & O'Hagan, 2001), (Williams, et al., 2006), and (Higdon, Gattiker, Williams, & Rightley, 2008) combine computer output with experimental data to determine the posterior distribution of the parameters that, based often on the mode of these distributions, are then chosen as the calibrated values.

Often, however, the best values for calibration parameters may be adjust as the value of the control inputs change (Atamturktur & Brown, 2015). These changing calibration parameters can then be

modeled as functions of the control input values using a Gaussian process that accounts for the functional uncertainty of the relationship (Brown & Atamturktur, 2018).

If a set of field measurements are made on a system at N different control input settings, the measured field data can be denoted as $y_i = y(x_i), i = 1, \dots, N$. The computer model to represent this system can be represented as $\eta(x, \boldsymbol{t})$ using the control input x and the vector of model parameters $\boldsymbol{t}$.

Thus, the following equation can be used to generate model output data corresponding to the measured field observations:

$$y_i = \eta(x_i, \boldsymbol{\theta}) + \varepsilon_i, \ i = 1, \dots, N \tag{4}$$

Where $\boldsymbol{\theta}$ is the vector of parameters that allow the model to exactly match the field measurements (the "true" parameters) and ε_i is a Gaussian normally distributed error term (with a variance λ_y that is also calibrated) for each control input to account for experimental measurement error.

In order to handle the case when $\boldsymbol{\theta}$ is a function of the control input, a Gaussian process model is used to represent the mapping between x and $\theta(x)$. All of the inputs are scaled to a unit hypercube and a Gaussian process correlation function is chosen that varies smoothly over the inputs and can be infinitely differentiable. Equation (5) below shows the chosen correlation function:

$$g\big(\theta_i(\cdot)\big) \sim GP\left(\mu_{\theta,i}, \ \lambda_{\theta,i}^{-1} \, R_i(\cdot,\cdot)\right); \quad R_i(x, x') = e^{\left\{-4\gamma_{\theta,i} \sum_{k=1}^{d_x} |x_k - x_k'|^2\right\}} \tag{5}$$

Where d_x the size of the control input vector is, $\lambda_{\theta,i}$ are the unknowns controlling the GP precision, $\gamma_{\theta,i}$ are the unknowns controlling the smoothness of the $\theta(\cdot)$ sample paths, and $\mu_{\theta,i}$ are the mean functions. This model is established once priors are placed on the GP hyperparameters. The smoothness parameter can be parameterized as $\rho_{\theta,i} = e^{-\gamma_{\theta,i}}$ and be assigned a Beta distribution prior of $\rho_{\theta,i} \sim Beta(1, b_\theta)$. The Beta parameter is chosen to place most density towards 1 in order to enforce *a priori* smoothness. Similarly, the precision parameter and error precision are assigned Gamma distribution priors of $\lambda_{\theta,i} \sim Ga(a_\theta, b_\theta)$ and $\lambda_y \sim Ga\big(a_y, b_y\big)$ and the parameters are chosen to again place most mass near 1 due to the fact that the calibration parameters are unit scaled.

The Markov chain Monte Carlo algorithm is used to simulate draws from the posterior distributions of the parameters. During this process, ρ_θ and θ are reparametrized on a log scale as $\nu = \log(-log(\rho_\theta))$ and $\xi = \log(-log(\theta))$, respectively, to remove the influence of the boundary constraints.

The functional parameter, $\boldsymbol{\theta}(x)$ is then sampled based on the current value of ρ_θ in order to take advantage of the smoothness assumptions. That is, to draw from the distribution $\boldsymbol{\theta}(x) \mid \xi, \nu, \lambda_\theta, \lambda_y, y$ the Cholesky decomposition of the covariance matrix is found and a proposal draw is made as $\boldsymbol{\theta}^*(x) = \boldsymbol{\theta}(x) + c \cdot \boldsymbol{B} \cdot \text{chol}(R_\nu)$ where $\boldsymbol{B} \sim N(0, \boldsymbol{I})$ and c is adaptive step size tuned during the burn-in iterations (and held constant during sampling) of the MCMC chain to maintain approximately 20-25% acceptance on the functional parameter and 40-50% acceptance on the constant parameters and ρ_θ. Sub-iterations are carried out to conduct additional draws and

tuning of $\boldsymbol{\theta}(x)$ for each total iteration of the remaining parameters. This is done to help improve the convergence of the functional parameter by taking more draws relative to the constant parameters and the GP hyperparameters.

3.1. Demonstration on an algebraic problem

The nonparametric calibration approach presented in this section is applied to the example algebraic problem from Section 2.1 after the sensitivity has revealed that τ should be treated as a functional parameter and θ_1 and θ_2 can be treated as constant parameters. Prior constraints of $2.35 \leq \theta_1 \leq 2.60$, $2.8 \leq \theta_2 \leq 3.3$, and $-0.5 \leq \tau \leq 2.5$ (a wide bounds to demonstrate the technique's ability to reduce uncertainty of the functional relationship) are placed on the parameters. The calibration routine is then carried out using 50,000 burn-in iterations and 30,000 sampling iterations, each with 15 sub-iterations on the τ vector. The posterior distributions of θ_1, θ_2, and τ are shown below in Figure 4 and Figure 5.

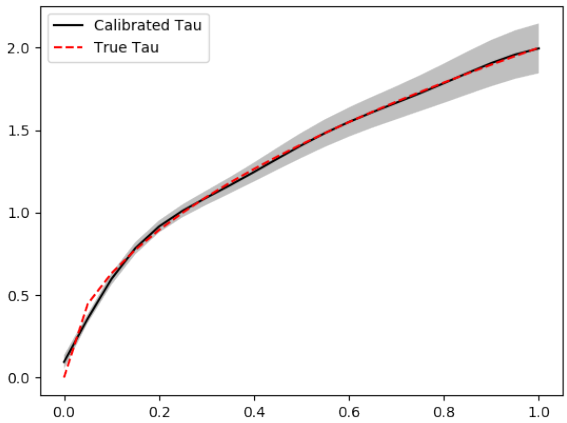

Figure 4. Calibrated function of τ over the domain x showing the pointwise posterior mean (black line) and one standard deviation bounds (grey region) compared to the true functional relationship (red dashed line).

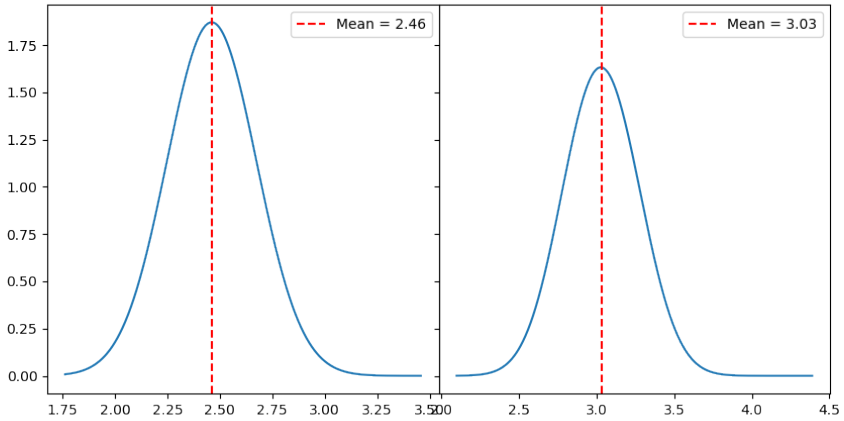

Figure 5. Posterior distributions of θ_1 (left; nominal value of 2.5) and θ_2 (right; nominal value of 3.0) after the nonparametric calibration technique.

Taking the mean values of θ_1 and θ_2 along with the τ vector based on the pointwise posterior means, the model is run and the resulting output is compared to the measured output. This comparison is shown below in Figure 6 and shows excellent agreement between the two responses throughout the entire control input domain.

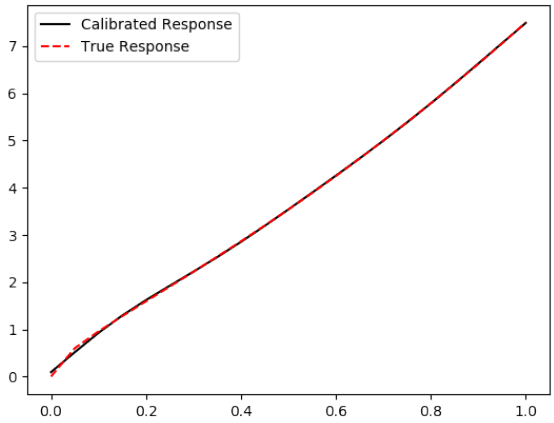

Figure 6. Plot of the model output using the posterior means of θ_1, θ_2, and τ compared to the actual measurements.

4. Example finite element problem

To demonstrate the nonparametric functional calibration method and functional sensitivity analysis on an actual physics problem, a finite element model was designed to simulate experimental measurements of a plate under tensile loading. The model was built such that the elastic modulus of the plate varies as a function of the plate temperature. This relationship between the elastic modulus and the temperature is assumed to be unknown to the model creator and is the functional relationship that the calibration method will seek to uncover about the material.

4.1. Model development

The finite element model is constructed using the Multiphysics Object-Oriented Simulation Environment (MOOSE) framework developed by Idaho National Laboratories (Gaston, Newman, Hansen, & Lebrun-Grandié, 2009). MOOSE provides a high-level interface to the PETSc nonlinear partial differential equation solver and the libMesh finite element library and is composed of several physics modules (tensor mechanics, heat conduction, contact, porous flow, and others) that can be coupled together as needed to solve complex problems. In addition, user-built applications can leverage the physics modules of MOOSE with additional physics for specific applications, such as BISON for light water reactor nuclear fuel performance calculations (Williamson, et al., 2012) and MOLTRES for molten salt reactor simulation (Lindsay, Ridley, Rykhlevskii, & Huff, 2018).

The specific model used in this example takes advantage of the tensor mechanics and heat conduction modules to represent the problem physics (Golchi, Bingham, Chipman, & Campbell, 2015). The plate is represented by a 10 by 10 element 2D mesh with constant specific heat and thermal conductivity and a modulus of elasticity that is a function of the nodal temperature. The

elasticity-temperature relationship of the material is shown below in Figure 7. The plate is then fixed on the left edge and a pressure loading of 2 MPa is applied to the right edge to place the plate in tension. In addition to the pressure loading applied to the edge of the plate, a 400K temperature gradient is generated over the part as shown below in Figure 8.

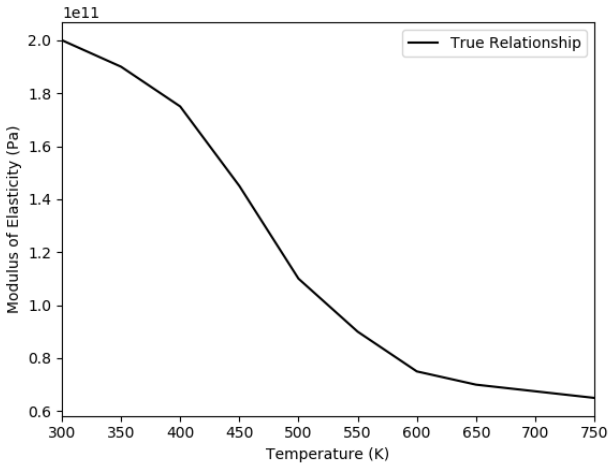

Figure 7. The relationship between the plate material's modulus of elasticity and the material temperature.

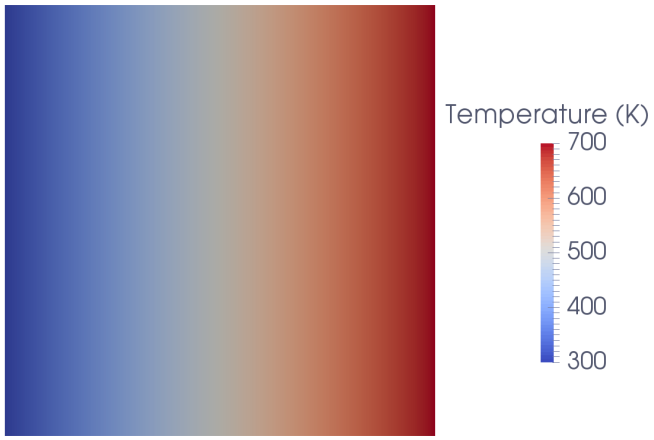

Figure 8. Illustration of the 400K temperature gradient across the plate.

To simulate the experimental measurement of the plate, synthetic displacement measurements were generated using the model under the established boundary conditions and loading. Figure 9 below shows a plot of the plate displacement and illustrates the discernable nonlinear nature of the curve due to the temperature dependence of the modulus of elasticity. The curve represents the experimental response that the finite element model should match after parameter calibration using the nonparametric calibration approach.

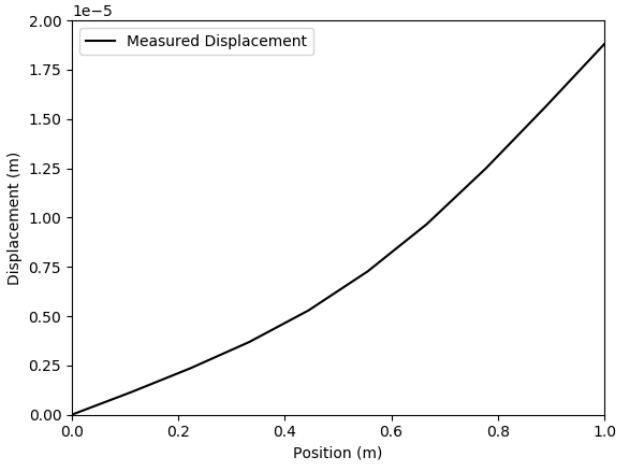

Figure 9. Plot of the synthetic displacement measurements generated by the finite element model.

4.2. Functional sensitivity analysis

In order to apply the functional sensitivity analysis of Section 2, the control input was specified as the temperature at the right boundary condition of the plate. Three different control input domains are chosen: 400K, 600K, and 800K. An additional calibration parameter is specified as the boundary condition temperature on the left end of the plate (set to 300K in the generated experimental measurements). This parameter was chosen because it would be reasonable to suggest that it may be dependent on the opposite boundary condition temperature and may be difficult to measure depending on the experimental setup. After the control inputs and parameters were specified, three Bayesian calibration runs were carried out for each of the input domains again using Sandia's Dakota software. The results are shown below in Table 3 and the relative variations are then calculated and shown in Figure 10.

Table 3. Results from calibration over each input range with the MOOSE FE model

Input Range	Temperature	Modulus
	$\bar{\mu}$	$\bar{\mu}$
800K	270.477	1.293E+11
600K	240.948	1.672E+11
400K	268.649	1.946E+11

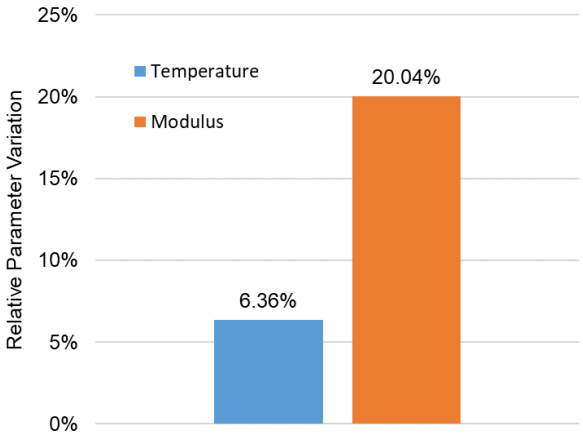

Figure 10. Relative variation calculations for the MOOSE model parameters.

Based on the results of this plot that the relative variation of the calibrated modulus of elasticity is over a 215% increase from the variation of the temperature boundary condition, it can be concluded that the modulus of elasticity would benefit much more from being treated as a functional parameter as was known to be the case. This is the parameter that will be used in the functional calibration approach presented in the following Section 4.3.

4.3. Nonparametric functional calibration

Once the finite element model has been created and the synthetic experimental data is available, the nonparametric calibration approach can be carried out on the computer simulation. In this problem, no assumptions were made about the functional dependence of the elasticity-temperature relationship ($\tau(T)$) other than maximum and minimum bounds over the function domain. The functional relationship was allowed to vary from a lower bound of 60 GPa to an upper bound of 220 GPa throughout the temperature domain in order to demonstrate the capability of this technique in the reduction of a very wide prior uncertainty about a functional relationship. In practice, however, if additional prior information about the functional relationship is known, it can be added to better inform the Gaussian process priors. Constraint options may include bounds on specific regions of the domain, such as a specified y-intercept range at the beginning of the domain, or trend information, such as enforcement that the function maintains monotonicity (Golchi, Bingham, Chipman, & Campbell, 2015). Without these types of more informed priors, the initial τ vector begins as an approximately horizontal line through the center of the domain at around 140 GPa.

Calibration was carried out on the MOOSE model using 800 burn-in iterations to refine the acceptance parameters and was followed 500 sampling iterations using constant acceptance parameters. During each iteration, 15 sub-iterations were conducted to make additional draws of the τ vector relative to the other parameters to help improve convergence of the posterior function. Figure 11 below show the posterior distribution of $\tau(T)$ with the calibrated relationship calculated as the pointwise mean of the vector with uncertainty bounds plotted as one standard deviation around the mean. The results show good agreement with the true relationship entirely contained within one standard deviation bounds.

It is interesting to note that the wider uncertainty bounds at the higher temperature bound can be explained by the lack of experimental data of the displacement response at temperatures above 700K. Additionally, at the lower temperature bounds the wider uncertainty is due to the relative insensitivity to temperature compared to the displacement occurring at the fixed end condition of the plate and the learned functional response deviates less from the initial value of 140 GPa than the true relationship. These points illustrate the importance understanding and applying informative priors on the functional relationship to ensure as accurate posterior results as possible over the entire control input domain.

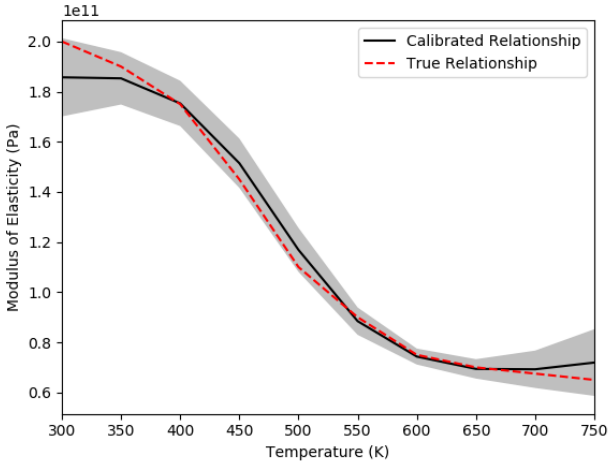

Figure 11. Posterior distribution of τ(x) with bounds of one standard deviation about the pointwise mean of the vector compared to the true material relationship.

Based on the calibrated mean functional relationship in Figure 11, the model simulation output is shown below in Figure 12. The model is also run with the upper and lower standard deviation bounds of τ and shown as the grey bounds in the plot. The calibrated output shows quite good agreement with the experimental displacement response throughout the plate and shows that the functional relationship in Figure 11 is an adequate representation of true functional relationship of the plate material to ensure model displacement output accuracy.

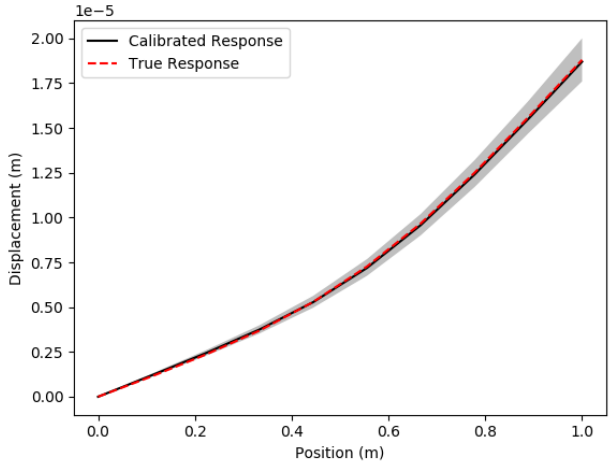

Figure 12. Plot of the displacement using the mean elasticity-temperature relationship compared to the synthetic experimental measurements with bounds based on the simulation results using the mean calibrated functional relationship plus and minus one standard deviation.

4.4. Comparison to non-functional calibration

As a way to verify the usefulness of this technique as compared to traditional scalar calibration, the calibration was repeated again to find the single most likely value of the material modulus of elasticity over the input domain. Again this calibration was run using Dakota and resulted in a posterior distribution for the modulus of elasticity with a mean value of 118.82 GPa. Figure 13 below shows the output of the MOOSE model when run with this constant modulus of elasticity compared to the results from the calibrated functional relationship. It is clear from this plot the improvement in simulated displacement response that the functional calibration gains over the single value calibration that essentially finds a modulus to best-fit the displacement measurements of the plate. The RMS error of the model output with functional calibration is a 99.53% reduction compared to the single value case (1.081E-09 versus 2.317E-07 when normalized by the average displacement). This is a drastic improvement in the simulation error and an example where the nonparametric functional approach would be invaluable to improving model accuracy.

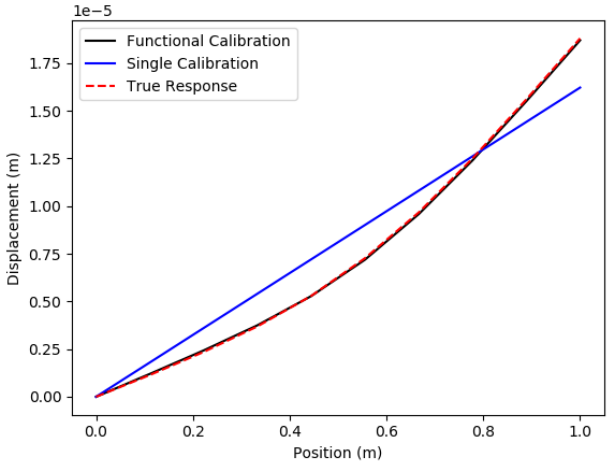

Figure 13. Plot showing the comparison between the functional calibration results and single parameter calibration.

5. Conclusions

The nonparametric functional calibration approach presented as a part of this framework provides a powerful way to improve physics-based model responses to match observed measurements where parameters have functional dependence on either an aspect of the operating domain or control inputs. The code used in this work has been written to flexibly interface with physics-based simulation codes using Python (easily adaptable to many different problems by adapting the system call to run the specific simulation code) and offers the ability to easily add more (or remove all) constant parameters to the calibration process. The code is inherently written to handle user adjustable settings for iterations, $\theta(x)$ sub-iterations, and parameter prior bounds. It also is built to read external text files containing the experimental measurements and automatically generate many of the results plots presented in this paper for analysis.

One area for future research in this work is in the formalizing of the sensitivity analysis method into a more quantitative indicator of parameter functional dependence. While visually it may be reasonable to determine which, if any, parameters would benefit from this approach, in practice it may be difficult to discern the results. Additionally, while in some problems (such as the algebraic problem in Section 2.1) it may be simple to determine which input is the independent control variable against which to conduct the analysis (there was only one), in the case of the finite element model, however, a single input (temperature) was chosen amongst many possible choices. This choice must be made through the use of expert opinion or the analysis can be repeated for every input that dependence is suspected. A method to expand upon this approach to determine which parameters are dependent and to what input inputs would make it much more applicable to a wide variety of complex engineering problems.

Overall, this work provides a solid foundation for a nonparametric functional calibration framework for physics models that has demonstrated its success in calibrating a multi-physics finite element model with a relatively common type of functional parameter that is encountered in

many engineering problems. The code is available to adapt easily to a wide range of simulation codes and additional, more informative prior constraints on the GP can be implemented into the base algorithm.

References

Adams, B. M., Bauman, L. E., Bohnhoff, W. J., Dalbey, K. R., Ebeida, M. S., Eddy, J. P., . . . and Wildey, T. M. (2014). "Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.0 User's Manual". *Sandia Technical Report SAND2014-4633*.

Atamturktur, S., & Brown, D. A. (2015). State-aware calibration for inferring systematic bias in computer models of complex systems. *NAFEMS World Congress Proceedings.* June 21-24, San Diego, CA.

Brown, D. A., & Atamturktur, S. (2018). Nonparametric Functional Calibration of Computer Models. *Statistica Sinica, 28*, 721-742.

Estacio-Hiroms, K. C., Prudencio, E. E., Malaya, N. P., Vohra, M., & McDougall, D. (2016). The QUESO Library, User's Manual. *Arxiv Preprint 1611.07521*.

Gaston, D., Newman, C., Hansen, G., & Lebrun-Grandié, D. (2009). MOOSE: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design, 239*(10), 1768-1778.

Golchi, S., Bingham, D. R., Chipman, H., & Campbell, D. A. (2015). Monotone Emulation of Computer Experiments. *SIAM/ASA J. Uncertainty Quantification, 3*(1), 370-392.

Higdon, D., Gattiker, J., Williams, B., & Rightley, M. (2008). Computer model calibration using high-dimensional output. *Journal of the American Statistical Association, 103*, 570-583.

Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J., & Ryne, R. D. (2004). Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing, 26*, 448-466.

Kennedy, M. C., & O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society, Series B*(63), 425-464.

Lindsay, A., Ridley, G., Rykhlevskii, A., & Huff, K. (2018). Introduction to Moltres: An application for simulation of Molten Salt Reactors. *Annals of Nuclear Energy, 114*, 530-540.

Williams, B., Higdon, D., Gattiker, J., Moore, L., McKay, M., & Keller-McNulty, S. (2006). Combining experimental data and computer simulation, with an application to flyer plate experiments. *Bayesian Analysis, 1*, 765-792.

Williamson, R. L., Hales, J. D., Novascone, S. R., Tonks, M. R., Gaston, D. R., Permann, C. J., & Martineau, R. C. (2012). Multidimensional multiphysics simulation of nuclear fuel behavior. *Journal of Nuclear Materials, 423*(1-3), 149-163.